DOI:
10.1007/978-3-642-34459-6_5

User-centric Principles in Automated Decision Making

Ingrid Nunes[1,2], Simon Miles[2], Michael Luck[2], and Carlos Lucena[1]

[1] PUC-Rio, Computer Science Department, LES - Rio de Janeiro, Brazil
{ionunes,lucena}@inf.puc-rio.br
[2] King's College London, Strand, London, WC2R 2LS, United Kingdom
{simon.miles,michael.luck}@kcl.ac.uk

Abstract. Natural-language preference expressions, not yet exploited by existing preference reasoning approaches, match the way users express preferences in many scenarios and potentially improve automated decision making. Further, the preferences provided are often not sufficient to make a choice on behalf of users, as trade-offs are resolved with psychological processes employed in light of available options. We thus propose a decision making technique that reasons about preferences expressed in a user-centric language and incorporates principles of trade-off contrast and extremeness aversion, as in human decision-making.

Keywords: Decision making, Preference Reasoning, Human Reasoning.

1 Introduction

Many everyday tasks involve decisions over a large number of options [10]: we must decide which clothes to wear, what to eat, where to go for fun. Both these regular decisions, and irregular ones, e.g. planning vacations, demand an effort that can be reduced by delegating decision-making to intelligent agents. For agents to appropriately perform tasks on our behalf, however, they must be aware of user preferences and the options available. While existing work allows agents to reason about a restricted set of preference types, this constrains users in how they express preferences and requires tedious interactive elicitation methods. We thus propose a novel approach to reasoning about preferences. Specifically, our contributions are: an automated *decision-making technique* based on preferences expressed in a *high-level preference language* and available options; exploiting principles regarding the way in which humans make choices; and an *evaluation* that compares decisions made by our technique with a human expert.

Our goal with the proposed technique is to simulate human reasoning in making decisions, allowing us to exploit natural user expressiveness of preferences (without the need for elicitation methods) and resolve trade-offs (that cannot be resolved with the provided preferences) in a way humans would do if provided with sufficient time and knowledge. We thus apply psychological processes used by humans, including mainly the principles of *trade-off contrast* and *extremeness aversion* [11]. Our decision-maker takes as input a set of options over which a choice is made, and a preferences expressed in a high-level language. It processes these to select one option, in such a way that the choice can be justified from the preferences. A decision here concerns choosing one option from a set *Opt* of the same conceptual class, e.g. apartments. Each class has a

Table 1: Preference Language (EBNF).

$preference$ ::= [$condition$] ($constraint$|$goal$|$order$|$qualifying$|$rating$|$indifference$|$dontCare$)
$condition$ ::= **if** $constraint$ **then**
$constraint$ ::= $expression$ | $expression$ **and** $expression$ | $expression$ **or** $expression$ | **not** $expression$
$expression$ ::= $attribute$ ($=$ | $\neq$ | $>$ | $\geq$ | $<$ | $\leq$) $value$
$goal$ ::= (**minimise** | **maximise**) $attribute$
$order$ ::= $attribute = value \succ attribute = value$
$qualifying$ ::= $performative$ $constraint$
$rating$ ::= $constraint$ $rate$
$indifference$ ::= **indifferent** $constraint$ {$constraint$}
$dontCare$ ::= **dont_care** $attribute$
$performative$ ::= [**dont**] (**prefer** | **need** | **desire** | **avoid** | **like** | **want** | **accept** | **require** | **love** | **hate**)
$rate$ ::= **best** | **v_good** | **good** | **neutral** | **bad** | **v_bad** | **worst**
$priority$ ::= [$condition$] ($attribute\_priority$ | $attribute\_indifference$ | $preference\_priority$)
$attribute\_priority$ ::= $attribute \rhd attribute$
$attribute\_indifference$ ::= $attribute \sim attribute$
$preference\_priority$ ::= $\mathbb{Z}$. $preference$

set Att of attributes, e.g. price, and each att_i is associated with a domain D_i that: (i) comprises a set of values x_{ij} allowed for the attribute; (ii) can be *discrete* or *continuous*; and (iii) can be *ordered* or *non-ordered*. As humans express preferences in many ways, we propose a preference language (Table 1), which is composed of different types of preferences and priorities and was derived from a previous study [9]. To illustrate, suppose *Bob* is visiting a university, and needs to choose an apartment to stay at. Each apartment is described in terms of four attributes: (i) distance from the university (**uni**); (ii) distance from the nearest underground station (**station**); (iii) chain (**chain**); and (iv) price per week (**price**). Bob's preferences are shown in Table 2a, with a prioritisation of attributes on the final line. The apartments available are in Table 2b.

Our technique initially analyses the options with regard to preferences, building two models for future use. As some preferences include important *implicit* information, in addition to their literal meaning, we extract this also. From the set of available options, we eliminate those that do not meet strict constraints, or are dominated in every regard by other options (though this latter step is not detailed in this paper due to space restrictions). As the remaining options have both costs and benefits, we take into account relative importance of attributes, and then go beyond the provided preferences with the user-centric principles, concluding with a decision. We make a few limiting assumptions: preferences are consistent (but may conflict); decisions do not concern *critical* matters, where a wrong choice may have serious consequences; decisions concern choice from a finite set of options; and, each preference (excluding its conditions) refers to only one attribute. The technique may be seen as a framework as it has variable parts, which were instantiated in this paper after running the technique with different alternatives, but our future work is to improve results by exploring this variability.

2 Models to Support Decision Making

2.1 Preference Satisfaction Model

Performatives such as *need* and *like* are widely adopted by users to express preferences over attributes, and so are included in our language. Similarly, users may rate preferences from best to worst. The relative importance of performatives, and their relation

1. **prefer** $uni \leq 2.5Km$
2. **if** $uni \leq 2.5Km$ **then need** $station \leq 1Km$
3. **if** $uni > 2.5Km$ **then need** $station \leq 0.7Km$
4. **minimise** $station$
5. **minimise** $price$
6. **avoid** $chain = D$
7. $chain = A \succ chain = B$
8. $chain = B \succ chain = C$

if $uni > 2.5Km$ **then** $station \rhd uni$

(a) Preferences.

Attribute	**Domain**	**Ap_A**	**Ap_B**	**Ap_E**
Uni (Km)	$\{x \mid x \in \mathbb{R}, x > 0\}$	5.5	2.0	3.5
Station (Km)	$\{x \mid x \in \mathbb{R}, x > 0\}$	0.40	0.45	0.65
Chain	$\{A, B, C, D\}$	C	D	A
Price ($)	$\{x \mid x \in \mathbb{R}, x > 0\}$	100	130	100

(b) Set of available apartments.

	uni	station	chain	price
Ap_A	¬ prefer	**need**	¬ avoid	
Ap_B	**prefer**	**need**	**avoid**	
Ap_E	¬ prefer	**need**	¬ avoid	

(c) PSM.

	Ap_A		**Ap_B**		**Ap_E**	
	Ap_B	**Ap_E**	**Ap_A**	**Ap_E**	**Ap_A**	**Ap_B**
uni	-	$\sim$	+	+	$\sim$	-
station	+	+	-	+	-	-
chain	+	-	-	-	+	+
price	+	$\sim$	-	-	$\sim$	+

(d) OAPM.

Table 2: Running example: preferences, options and decision models.

to rates, is specific to each individual, and eliciting this is outside the scope of this paper. Here, we adopt one ranking and categorisation as positive, negative and neutral, shown in Figure 1. Rates and performatives (collectively referred to as *modifiers*, M) used in preferences are captured by a *Preference Satisfaction Model (PSM)*, a table indicating how options satisfy preferences in terms of each attribute. This maps a pair $\langle option, attribute\rangle$ to a performative or rate (or their negation): $PSM : Opt \times Att \rightarrow \{empty, \neg\} \times M$. Modifiers come from qualifying and rating preferences but also constraints, interpreted as the performative "*want.*". Thus, each preference considered in the PSM consists of a modifier, a constraint, and, optionally, a condition.

The PSM is constructed as follows. For each preference, the relevant attribute of each option is tested to see if the preference is satisfied (condition and constraint met). For each option-attribute pair, the modifier from one preference is chosen as the respective value in the PSM as follows. If at least one constraint is satisfied, the strongest modifier according to our scale is chosen, either the most positive or most negative (depending on whether the modifier is a positive or negative one). If no statement is satisfied, the least negative modifier is chosen, and marked negated in the PSM. Table 2c shows the PSM for our running example.

2.2 Options-Attribute Preference Model

To capture comparisons between options, we introduce the Options-Attribute Preference Model (OAPM), a table that, for individual attributes, shows which of each pair

+	`require > need > love/want/best > desire/v_good > prefer/like/good > accept`
+-	`don't require > neutral/don't love/don't hate > dont't need/don't avoid/don't desire`
-	`don't prefer > avoid/bad > don't like/v_bad > don't want/worst > hate > don't accept`

Fig. 1: Modifier strength scale (performatives and rates).

of options is better. Four possible values relate each option A to an option B w.r.t. an attribute X: (i) (+) the attribute value of A is better than B; (ii) $(-)$ A is worse than B; (iii) $(\sim)$ A is as preferred as B; and (iv) (?) no conclusion can be drawn with the provided preferences. This is represented as $OAPM : Opt \times Opt \times Att \rightarrow \{+, -, \sim, ?\}$. The result for our running example is shown in Table 2d. The OAPM is initialised with all values set to "?". We then analyse the following in order: (i) the PSM; (ii) goals; (iii) order and indifference preferences. The ordering allows each step to override or refine values derived from prior steps.

PSM. We use our modifier scale (Figure 1) to determine the preferred value of two options' attributes, following the rules specified below, shown as situations in which the PSM establishes preference of o_1 over o_2. Note that an absence of satisfied preferences for an option, $\langle \neg, modifier \rangle$, is considered an indicator that the option is undesirable, following typical practice that people explicitly state where attribute values are desired, acceptable, not to be avoided etc. [10]. In our running example, $OAPM(Ap\_A, Ap\_B, uni)$ is set to $-$, as $PSM(Ap\_A, uni) = \neg prefer$ and $PSM(Ap\_B, uni) = prefer$.

$\mathbf{o_1}$	$\langle empty, stronger\ +\ m \rangle$	$\langle empty, +m \rangle$	$\langle empty, \pm m \rangle$	$\langle empty, \pm m \rangle$	$\langle \neg, m \rangle$
$\mathbf{o_2}$	$\langle empty, m \rangle$	$\langle \neg, m \rangle$	$\langle \neg, +m \rangle$	$\langle \neg, \pm m \rangle$	$\langle empty, -m \rangle$

Goals. When there is a goal to maximise the attribute, the option with a higher value is set to $+$, and the other set to $-$. The inverse applies for a minimisation goal. According to the PSM, Ap_A and Ap_E are similar w.r.t. $station$ (both are associated with $\langle empty, need \rangle$), but as there is a goal to minimise this attribute, the OAPM is updated to $OAPM(Ap\_A, Ap\_E, station) = +$ and $OAPM(Ap\_E, Ap\_A, station) = -$.

Order preferences. Order preferences state a partial order between values of an attribute, where different orders may apply under different conditions. We create a graph, an *attribute value partial order (AVPO)*, for each option-attribute pair, whose nodes contain equality expressions of order preferences (which represent domain values), while directed edges denote the preference of one value to another. For example, preferences 7 and 8 of our running example results in the following AVPO: $chain = A \rightarrow chain = B \rightarrow chain = C$ — for all options, as there are no conditions. For a given attribute and options o_1 and o_2, where there is a path from the node that satisfies o_1's value to that satisfying o_2's value in both options' AVPO, then this means the orders applicable to each option both consider o_1 preferable to o_2 for that attribute, and the OAPM entries are set to $+$ and $-$ accordingly. This is the case with options Ap_E and Ap_A, whose chains are A and C, respectively.

Indifference preferences. The OAPM value of an option-attribute pair is set to $\sim$ if the attribute values satisfy at least one constraint of the same indifference preference, and its condition (if any) is satisfied by both options.

Preferences always provide a literal meaning, but can also bring additional information to derive new preferences, referred to as *implicit preferences*. These never override explicitly provided preferences, but aid ordering attribute values when this is otherwise inconclusive, i.e. where $OAPM(o_1, o_2, att) = \sim \vee\ ?$ and this is not due to an explicit indifference preference. We consider four kinds of implicit preference in refining the OAPM. First, when an **upper bound** is specified for an attribute, we assume that this implies a goal to minimise the attribute value. For example, a user expecting to pay at most \$100 for a hotel night also wants to minimise price. Conversely, a **lower bound** implies a goal to maximise the attribute value. A **reference value (around preference)** implies a goal that a value closer to the reference is better. Finally, an **interval** of acceptable values ($att > lowerLimit \wedge att < upperLimit$) implies, for values outside that interval, a goal of being closer to the nearest interval boundary. For all such implicit preferences, if the modifier associated with the qualifying or rating preference is negative, the effect is inverted. In our example, preference 1 suggests an implicit preference for *minimising* the value of the uni attribute.

3 Cost and Benefit Analysis

We next assess the relative costs and benefits of pairs of options. We first analyse the benefits of option o_1 compared to option o_2 for each attribute, and the same for o_2 compared to o_1. The benefits of option o_1 w.r.t. o_2 are the costs of option o_2 w.r.t. o_1. $Benefit : Opt \times Opt \times Att \rightarrow \{x | x \in \mathbb{R} \wedge 0 \geq x \geq 1\}$ indicates how much better one option is than another, w.r.t. to one attribute. If the $OAPM(o_1, o_2, att) \neq +$, then the benefit is 0, otherwise, $Benefit(o_1, o_2, att)$ is computed based on the *reason*, preference or PSM value, for setting the final OAPM value as described above. Our decision maker keeps track of these reasons when building the OAPM. Different cases are considered depending on the reason.

First, the reason may be a goal or an implied preference of a kind described above. If a **goal, upper or lower bound**, the benefit is the difference between the options' attribute values. If an **around preference**, the benefit is the difference between the distances of options' values from the reference value. If an **interval preference**, the benefit is the difference between the distances of options' values from the nearest interval boundary. In all these cases, the difference is normalised to $[0, 1]$ within the minimum and maximum domain value bounds of the attribute (which are given, or we extract from the the available options).

If the reason is **PSM values**, we assess how much one value is preferred to another. We order our modifiers, and associate a numeric value with each level of the scale, with 0 for the middle level (*neutral*, *don't love*, *don't hate*), increasing 1 for each level above and reducing by 1 for each level below. The absolute benefit for PSM value $\langle n, m \rangle$ with modifier m at level $level$ is as follows: $f_m(level) = \log(|level| + 1)$, if $level \geq 0$, $n = empty$; $f_m(level) = -\log(|level| + 1)$, if $level < 0$, $n = empty$; and $f_m(level) = 0, n = \neg$. The relative benefit between options is then the difference between the absolute benefits of the options. We use a logarithmic function above so that differences between stronger modifiers, such as *require*, are less than differences between modifiers in the middle of the scale, such as *don't avoid*, consequently the preference is much stronger when comparing positive modifiers with negative modifiers.

If the reason is an **order preference**, then the AVPO graphs are used to calculate the benefit. Each AVPO node, corresponding to an attribute value, is tagged with a modifier according to the same algorithm used to construct the PSM for the options, i.e. whether that value is preferred, not avoided etc. Nodes with only incoming or outgoing edges that cannot be tagged are given default modifiers, *want* and *prefer* respectively, or stronger modifiers if there are other tagged nodes with more positive or negative

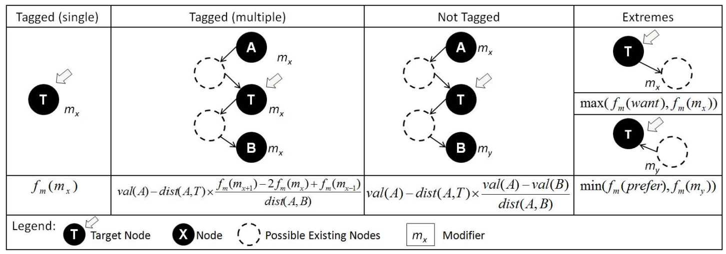


Fig. 2: Calculating order node values.

modifiers than the default. Less preferred nodes are tagged with *prefer* by default, since people typically provide an order for preferred or acceptable values, and ignore others. Each node is then given absolute benefit value as follows (some details omitted due to space restrictions). If a modifier is tagged to only one node, then its benefit is calculated using f_m above. If multiple nodes have the same modifier, with level $level$, their benefits are an even distribution from $f_m(level) - (f_m(level) - f_m(level-1)/2)$ to $f_m(level) + (f_m(level+1) - f_m(level)/2)$ following the edge order in the AVPO. Untagged nodes' benefits are determined from those of connected nodes. Figure 2 illustrates the calculation. The relative benefits between options are then calculated as the difference between absolute benefits of the nodes containing the two options' values in each AVPO, normalised to [0,1] by the minimum and maximum values of the modifier scale. The values from the two AVPOs are then averaged to reach the final benefit value. Nodes of the $chain$ AVPO of our example are tagged with $\neg avoid$, and their values are calculated with the equation of multiple tagged nodes.

Benefits of options across all attributes are calculated by considering the prioritisation of preferences and attributes expressed with the priorities. First, for each option, the preference priority order (1 to 8 in our example) gives an initial ordering of attributes: attributes constrained by higher priority preferences whose conditions hold for that option are more important, e.g. Ap_A gives order $uni \succ station \succ price \succ chain$. Next, we consider those explicit attribute prioritisations and indifference whose conditions hold, and change the order accordingly, e.g. for Ap_A, $station$ and uni are swapped. Finally, don't care preferences indicate attributes to be removed from the order. Given this attribute order, we take the least important attributes to be level 1 in the order, and the longest path in the order from the least important attributes to the most important ones is referred to as $size(attO)$. We use a logarithmic function ($f_a(x) = a \log x + b$) to calculate attribute weights when considering the overall option benefits, with $f_a(1) = 1$ and $f_a(size(attO)) = size(attO)$. We then calculate parameters a and b, according to the number of levels. The logarithmic function, with characteristics imposed by the points established, gives a much higher priority to more important attributes, which have similar importance (in comparison to a linear function). Based on the logarithmic function with specific parameters, $w_i = f_a(level(i)) / \sum_{j=1}^{size(att)} f_a(level(j))$ calculates the weight of each attribute $w_i \in Att$. Now that we have the benefits of an option o_1 w.r.t. an option o_2, for each individual attribute, and also its weights, we calculate the overall benefits from o_1 w.r.t. o_2 using a weighted sum.

3.1 Taking into Account User-centric Principles

As we are not considering dominated options in this paper, options have both pros and cons, and thus a trade-off must be resolved to choose one of them. People not only consider the two options being compared, and their costs and benefits, but also the cost-benefit relationship (ratio), which is positioned in relation to this ratio between other options [11]. This is referred to as *trade-off contrast*. We therefore incorporate a new factor in the process of choosing an option, based on a function that shows the trade-off between two options TO. This is a partial function defined as $b(o_2, o_1)/b(o_1, o_2)$, whose domain is every pair of different options for which $b(o_1, o_2) > b(o_2, o_1)$. As the function b is always a value in the interval $[0, 1]$, the TO function is always a value in this interval, excluding its boundaries. The trade-off between two options is not isolated; with only two options, all we know is that one option has more benefits than another. When there are other options, and the (human) decision-maker observes that the cost-benefit relationship is better for other options, they see it as a negative aspect of the option. This counters *rational* decision-making, as preferences for a particular option do not depend on available options. Based on the TO function, the option benefits w.r.t. trade-off are as below, having as a basis the avg_{TO} (average of all values defined for the TO), which determines when the trade-off is a benefit or a cost.

$$to(o_1, o_2) = \begin{cases} avg_{TO} - TO(o_1, o_2) & \text{if } TO(o_1, o_2) \text{ is defined and } TO(o_1, o_2) \le avg_{TO} \\ TO(o_2, o_1) - avg_{TO} & \text{if } TO(o_2, o_1) \text{ is defined and } TO(o_2, o_1) > avg_{TO} \\ 0 & \text{otherwise} \end{cases}$$

Humans also consider how *extreme* options are. Extreme options are close to best for some attributes, e.g. quality, but incur a high penalty for others, e.g. price. In general, humans avoid extreme options [11], referred to as *extremeness aversion*. To evaluate how extreme options are, we calculate the distance between an option's attribute values from the best possible value according to preferences whose conditions are valid for the option ($bestDist(o, att)$). The precedence order for using preferences to calculate the distance from best is: (i) goals and implicit preferences, (ii) order, and (iii) PSM. An attribute is considered only if there is no *don't care* preference associated with it. Extreme options have low costs for some attributes ($bestDist$ close to 0) and high costs for others ($bestDist$ close to 1), so extremeness of an option is given by the standard deviation of the function $bestDist$ for a particular option: $ext(o) = STDEV(\{bestDist(o, att_i)|i = 1...size(att)\})$, which is a value between 0 and 1. Finally, extremeness aversion suggests that a less extreme option has a benefit with respect to a more extreme option. To capture this aspect, we define the function $ea(o_1, o_2)$, which is $ext(o_2) - ext(o_1)$ if $ext(o_1) < ext(o_2)$, and 0 otherwise.

3.2 Comparing Option Relative Values

We have analysed three aspects of options: benefits, trade-off relative to available options, and extremeness. The last two aspects are also seen as benefits: trade-offs better than the average are also a benefit, and the the least extreme of two options has a benefit w.r.t. the other. The final value $v(o_1, o_2)$ of an option is a weighted sum of these aspects — we are currently using default weights of 0.25 for trade-off contrast and 0.15 for extremeness aversion. We identify the chosen option as better than or equal

Table 3: Running Example: choosing an option.

	uni	$station$	$chain$	$price$	$b(o_1, o_2)$	$ea(o_1, o_2)$	$TO(o_1, o_2)$	$to(o_1, o_2)$	$v(o_1, o_2)$	**Balance**
Ap_A	0.0	0.031	0.031	0.230	0.291	0.227	0.421	0.320	0.289	**0.216**
Ap_B	0.123	0.0	0.0	0.0	0.123	0.0		0.0	0.073	
Ap_B	0.123	0.103	0.0	0.0	0.226	0.0		0.126	0.167	
Ap_E	0.0	0.0	0.031	0.230	0.260	0.144	0.868	0.0	0.178	**0.011**
Ap_A	0.0	0.153	0.0	0.0	0.153	0.084		0.194	0.153	**0.055**
Ap_E	0.124	0.0	0.040	0.0	0.164	0.0	0.935	0.0	0.098	

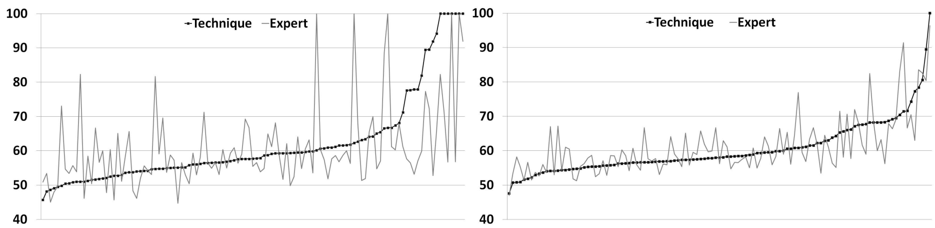
to ($v(o_1, o_2) \geq v(o_2, o_1)$) every other option. If different options have the same value w.r.t. another ($v(o_1, o_2) = v(o_2, o_1)$), and they are better than every other option, we randomly choose one of them. Cycles may arise in the v function if different criteria are used to compare attribute values of each option pair; e.g., the price of o_1 and o_2 is compared by goal, and of o_1 and o_3 by constraint. As preferences are consistent, price alone gives no cycles; however, as v is calculated for different preferences, differences in scales can lead to cycles when considering overall option benefits. To choose one option in this situation, from the set of options that are considered better than the highest number of options, we choose that with the minimum of the maximum balances for every option that is considered better than it. In our experiments, on real user data, there were only 16 (of 113) occurrences of cycles. Even though this number is low and we use a workaround to solve this issue, it is future work to completely eliminate cycles.

In our example, our technique results in the values presented in Table 3. Here, the **chosen option is Ap_A**, which would not have been chosen without our user-centric principles. As already introduced, our technique goes beyond provided preferences, because they do not give enough information to resolve trade-offs, which humans do *during* decision-making. Our technique aims to anticipate this preference construction in order to make a decision on behalf of the user or provide a recommendation.

4 User Study-based Evaluation

We evaluate our decision-maker by empirical evaluation but, as the input of our technique is high-level preferences and existing approaches cannot handle all of them, we restrict ourselves to making a side-by-side comparison with a human expert. The evaluation is based on the study [9] that also informed the preference language itself. Participants provided preference specifications (in natural language) for use by an individual to buy a laptop on their behalf. Both the participants and domain expert (based on the participant's preferences), were given a catalogue with 144 laptops from which to choose up to five options. We compared decisions made by our technique based on provided preferences against those of the user and expert. Similarly to how the expert recommendation was assessed in the user study, we calculate a similarity score SS (which ranges from 0 to 100), comparing the recommendation with the user choice and taking into account the position of the up to five chosen laptops using a weighted average.

We run our technique with 113 preference specifications — keeping only keywords to be consistent with our language — as input (taking an average of 10.2832 $seconds$ on an Intel Core 2 Quad $2.66GHz$, $4GB$ of RAM, with standard deviation 0.6465,



(a) First choices.

(b) Up to five choices.

Fig. 3: Expert vs technique.

Table 4: Reasoning Approaches vs. Preferences.

Approach	Preference								Attribute	Attribute	Preference
	Condition	Const.	Goal	Order	Qualif.	Rating	Indiff.	Don't care	Priority	Indiff.	Priority
UF-based [8]				X							
SVM-based [6]				X		X	X				
Soft Constraints [2]		X									X
CP-Nets [3]	X			X							
TCP-Nets [4]	X			X					X		
Scoring Function [1]		X				X		X			X
Winnow [5]		X		X			X				X

to be executed for each request, with 144 laptops, and 61 attributes), and obtained the similarity scores shown in Figure 3a (first expert and technique choices compared to the first user choice) and 3b (first up to five expert and technique choices compared to the first up to five user choices). Specifically, in the first choice comparison, the average SS of our technique is 61.94 ($stdev = 13.24$, $min = 45.69$, $max = 100.0$), while of the expert is 60.79 ($stdev = 12.05$, $min = 44.72$, $max = 100.0$). And in the up to five choices comparison, the average SS of our technique is 60.36 ($stdev = 7.69$, $min = 47.56$, $max = 100.0$), while of the expert is 61.03 ($stdev = 8.38$, $min = 46.97$, $max = 96.39$). The small difference in obtained values is not statistically significant, as determined by a paired-samples t-test: $t(0.916) = 0.361, p = 0.05$ (*F*) and $t(-1.268) = 0.207, p = 0.05$ (*5*). Therefore, we cannot reject the null hypothesis that domain expert and technique choices are different, and *we can conclude that our technique makes choices as good as those of the domain expert.*

5 Related Work and Conclusions

Most existing work related to decision making is founded on Multi-Attribute Utility Theory [7], which emphasises the use of multi-attribute preference models based on *utility functions* (UFs). Many approaches [8, 6] propose specific models to represent preferences for deriving *utility functions*. Some approaches [2] extend Constraint Satisfaction Problems (CSPs) to incorporate *soft constraints* (that can remain unsatisfied), associating preference with each constraint, and creating an optimisation problem of maximising preference. UFs and CSPs are classical approaches for dealing with preferences and making decisions, but the former are hard to elicit, and the latter deal with over-constrained problems rather than choosing from feasible solutions. A third group of approaches [3, 4] proposes new *graphical structures* to represent and reason about preferences. Finally, work in the area of databases proposes *extensions of query languages* [1, 5] to incorporate preferences and algorithms to provide query results according to specified preferences. Even though these approaches propose different solutions, they share the common goal of making a preference-based choice. However, they address limited kinds of preferences (Table 4), restricting human natural expression, and cannot make a decision when the preferences themselves do not lead to a single option to be chosen. As this paper is not concerned with preference elicitation methods, they have not been considered.

In this paper, we provided an approach to reasoning about preferences and making decisions. Our technique provides the novelty of exploiting different natural language expressions and user-centric principles in automated decision making, and these can be used as general lessons in this research area: performatives and other expressions give valuable information that can be used to generate low-level preference representations (such as utility functions), and these (and possibly others) user-centric principles can be used to reduce the amount of preferences obtained from users, as they can predict how users would resolve trade-offs. Moreover, these principles of human decision making explain situations in which a decision made by a human is “irrational” according to classical decision theory, and by taking these principles into account, automated systems can make decisions that are more acceptable to users. Short-term future work is to explore variable parts of our technique, such as modifier interpretation, modifier and weight functions, and weights used for trade-off contrast and extremeness aversion.

References

1. Agrawal, R., Wimmers, E.L.: A framework for expressing and combining preferences. In: Proceedings of the 2000 ACM COMAD. pp. 297–306. ACM (2000)
2. Bistarelli, S., Montanari, U., Rossi, F.: Semiring-based constraint satisfaction and optimization. J. ACM 44, 201–236 (1997)
3. Boutilier, C., Brafman, R.I., Domshlak, C., Hoos, H.H., Poole, D.: Cp-nets: a tool for representing and reasoning with conditional ceteris paribus preference statements. J. Artif. Int. Res. 21(1), 135–191 (2004)
4. Brafman, R.I., Domshlak, C., Shimony, S.E.: On graphical modeling of preference and importance. J. Artif. Int. Res. 25, 389–424 (2006)
5. Chomicki, J.: Preference formulas in relational queries. ACM Trans. Database Syst. 28, 427–466 (2003)
6. Domshlak, C., Joachims, T.: Efficient and non-parametric reasoning over user preferences. User Modeling and User-Adapted Interaction 17(1-2), 41–69 (2007)
7. Keeney, R.L., Raiffa, H.: Decisions with Multiple Objectives: Preferences and Value Tradeoffs. Wiley series in probability and mathematical statistics, Wiley (1976)
8. McGeachie, M., Doyle, J.: Utility functions for ceteris paribus preferences. Computational Intelligence 20(2), 158–217 (2004)
9. Nunes, I., Barbosa, S., Lucena, C.: Understanding how users express preferences: a user study. Tech. Rep. CS-2010-19, University of Waterloo, Canada (2010)
10. Schwartz, B.: The paradox of choice: Why more is less. Harper Perennial (2005)
11. Simonson, I., Tversky, A.: Choice in context: Tradeoff contrast and extremeness aversion. Journal of Marketing Research 29(3), 281–295 (1992)